最小ハミング距離法のファイル構成への応用 (実験配置の理論と応用)

AUTHOR(S):

高橋, 磐郎

CITATION:

高橋, 磐郎. 最小ハミング距離法のファイル構成への応用 (実験配置の理論と応用). 数理解析研究所講究録 1980, 404: 90-100

ISSUE DATE:

1980-11

URL:

http://hdl.handle.net/2433/102313

数理解析研究所講究録
404巻 1980年 90-100

最小ハミング距離法のファイル構成への応用

筑波大学社会工学系
高橋磐郎

§0 序

リレーショナルデータベース等におけるファイルは，表2（の$x_1 \sim x_7$列）のようなものと考えられる。これは$m=7$項目，各項目のカテゴリ数$q=2$，レコード数$n=16$（$\leq q^k = 2^4$）のものである．qが素数べきのとき，各項目のカテゴリを$GF(q)$の元と考えると，ファイルRは$GF(q)^m$の大きさnの部分集合である．

Bose等は，$GF(q)^m$の中に強さtのk次元線形部分空間S（強さtの直交表）を考え，Sの各点をブロック（バケットの集り）とみなすファイルによって，t項目質問に答え得るファイリングシステムを提案し，これが普通の転置方式より冗長度が少いことを証明している。しかしこの方法はRの点の分布に無関係にSを構成しているため，Rの点の分布が一様性をもつ場合にしか，その有用性が謳われない．実際に

Rの点は $GF(q)^n$ の中にきわめて疎にしかもかなり偏って分布している. [1][2]

ここでは与えられたRにある意味で最もよく適合する線形部分空間Sにもとづくファイリングシステムを提案する. 与えられたRに対してSを求めるのに, 実数の世界での最小二乗法に匹敵する, 最小ハミング距離法なるものを利用する. このアルゴリズムは本質的に, コード理論における リードマラーコードのデコード方式と同等のものである. [0]

§1 最小ハミング距離法によるあてはめ

$GF(q)(=GF(2))$ 上の $k(=3)$ 変数関数 $y=f(x_1, x_2, x_3)$ の値が表1のように与えられたとき, これに1次式

(1) $y_\nu = a_0 + a_1 x_{\nu 1} + a_2 x_{\nu 2} + a_3 x_{\nu 3} + e_\nu$

$\nu = 1, \cdots, q^k = n, \quad a_i \in GF(q)$

をあてはめる問題を考える. その規則は誤差ベクトル $\mathbf{e} = (e_\nu : \nu = 1, \cdots, n)$ のハミングウェイトが最小になるように, つまり $\mathbf{y} = (y_\nu : \nu = 1, \cdots, n)$ と $(a_0 + a_1 x_{\nu 1} + a_2 x_{\nu 2} + a_3 x_{\nu 3} : \nu = 1, \cdots, n)$ とのハミ

表 1

ν	x_1	x_2	x_3	$y=f(x_1,x_2,x_3)$
1	0	0	0	0
2	0	0	1	1
3	0	1	0	0
4	0	1	1	1
5	1	0	0	1
6	1	0	1	1
7	1	1	0	1
8	1	1	1	0

ング距離が最小になるように, a_0, a_1, a_2, a_3 をきめるものとする.

このような a_i を最小ハミング距離推定 $\hat{a}_i$ と呼ぶとすると, GF(2) の場は, これはつぎのような式で決められることがわかる.

$$(2)\quad \begin{cases} \hat{a}_1 = \mathrm{maj}\{\sum_{x_1 \in GF(2)} f(x_1,0,0),\ \sum_{x_1} f(x_1,0,1),\ \sum_{x_1} f(x_1,1,0),\ \sum_{x_2} f(x_1,1,1)\} \\ \hat{a}_2 = \mathrm{maj}\{\sum_{x_2} f(0,x_2,0),\ \sum_{x_2} f(0,x_2,1),\ \sum_{x_2} f(1,x_2,0),\ \sum_{x_2} f(1,x_2,1)\} \\ \hat{a}_3 = \mathrm{maj}\{\sum_{x_3} f(0,0,x_3),\ \sum_{x_3} f(0,1,x_3),\ \sum_{x_3} f(1,0,x_3),\ \sum_{x_3} f(1,1,x_3)\} \end{cases}$$

$$(3)\quad \hat{a}_0 = \mathrm{maj}\{y'_\nu \mid \nu = 1, \cdots, n\},$$

$$y'_\nu = y_\nu - \hat{a}_1 x_{\nu 1} - \hat{a}_2 x_{\nu 2} - \hat{a}_3 x_{\nu 3},\quad \nu = 1, \cdots, n$$

ここで maj は多数決関数, つまり $\mathrm{maj}\{0,0,1\} = 0$, $\mathrm{maj}\{1,0,1\} = 1$ など. tie が起る場合および $d(\boldsymbol{y}, \hat{\boldsymbol{y}}) > 2^{m-2}$ の場合は [4] 参照. ただし $\hat{\boldsymbol{y}} = \hat{a}_0 + \hat{a}_1 \boldsymbol{x}_1 + \hat{a}_2 \boldsymbol{x}_2 + \hat{a}_3 \boldsymbol{x}_3$ $(\boldsymbol{x}_i = (x_{\nu i} : \nu = 1, \cdots, n))$ で $d(\boldsymbol{y}, \hat{\boldsymbol{y}})$ は $\boldsymbol{y}$ と $\hat{\boldsymbol{y}}$ とのハミング距離.

表1に対して (2), (3) を適用すると,

$$(4)\quad \hat{\boldsymbol{y}} = \boldsymbol{x}_1 + \boldsymbol{x}_3, \qquad d(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1$$

である. この公式は本質的には, 1次のリードマラーコード

における デコード方式と同等である．一般に k 次式のあてはめも同様な方法で可能であり，これは k 次のリードマラーコードの場合に対応する[0]．

§2 リレーショナルファイルにあてはめられた線形部分空間

表2（$x_1 \sim x_7$ 列）のようなファイル R が与えられているとしよう．これにキー（t_1, t_2, t_3, t_4）（表2 $t_1 \sim t_4$ 列）を対応させ，各 x_i を t_1, t_2, t_3, t_4 の関数とみて最小ハミング距離法で1次式のあてはめを行うと，つぎのようになる

$$
(5)\quad \begin{cases}
\hat{x}_1 = t_1 + t_2 & (2)\\
\hat{x}_2 = t_2 + t_3 & (2)\\
\hat{x}_3 = t_2 + t_4 & (2)\\
\hat{x}_4 = t_1 & (2)\\
\hat{x}_5 = t_1 + t_3 & (1)\\
\hat{x}_6 = t_2 & (2)\\
\hat{x}_7 = t_1 + t_3 + t_4 & (2)
\end{cases}
$$

（カッコ内はあてはめの誤差 $d(x_i, \hat{x}_i)$ と示す）．

これは $GF(q)^m = GF(2)^7$ の $k=4$ 次元線形部分空間 S を表現する1次式である．S の各点がブロックあるいはバケットでありこれを表3に示す．

表2（R）

record	t_1	t_2	t_3	t_4	x_1	x_2	x_3	x_4	x_5	x_6	x_7
⓪	0	0	0	0	0	1	0	0	0	0	0
①	0	0	0	1	0	0	1	0	0	0	1
②	0	0	1	0	1	1	0	0	1	0	1
③	0	0	1	1	0	1	1	0	1	0	1
④	0	1	0	0	1	1	1	1	0	1	0
⑤	0	1	0	1	1	1	0	0	0	1	1
⑥	0	1	1	0	1	1	1	0	1	1	1
⑦	0	1	1	1	1	0	0	0	0	0	0
⑧	1	0	0	0	1	0	1	1	1	0	0
⑨	1	0	0	1	0	0	0	1	1	0	0
⑩	1	0	1	0	1	1	0	1	0	1	0
⑪	1	0	1	1	1	1	1	1	0	0	1
⑫	1	1	0	0	0	1	1	0	1	1	1
⑬	1	1	0	1	0	1	0	1	1	1	0
⑭	1	1	1	0	0	0	1	1	0	1	0
⑮	1	1	1	1	0	0	0	1	0	1	1

表3（S）

$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$	$\hat{x}_6$	$\hat{x}_7$	stored record
0	0	0	0	0	0	0	⓪ ① ② ③
0	0	1	0	0	0	1	① ③
0	1	0	0	1	0	1	⓪ ① ② ③
0	1	1	0	1	0	0	⓪ ③
1	1	1	0	0	1	0	⓪ ① ② ③
1	1	0	0	0	1	1	②
1	0	1	0	1	1	1	① ② ③
1	0	0	0	1	1	0	②
1	0	0	1	1	0	1	① ② ③
1	0	1	1	1	0	0	② ③
1	1	0	1	0	0	0	⓪ ②
1	1	1	1	0	0	1	② ③
0	1	1	1	1	1	1	⓪ ① ② ③
0	1	0	1	1	1	0	⓪
0	0	1	1	0	1	0	①
0	0	0	1	0	1	1	

§3 ストアアルゴリズム

(5)の一般形を

(6) $\hat{x}_i = b_{i1}t_1 + \cdots + b_{ik}t_k \quad , i=1,\cdots,m$

としておくと，レコード $x=(x_1,\cdots,x_m)$ をキー $t_\nu=(t_{\nu 1},\cdots,t_{\nu k})$ に対応するバケット $\hat{x}_\nu=(\hat{x}_{\nu 1},\cdots,\hat{x}_{\nu m})$ にストアするか否かを決めるアルゴリズムは；

(7) $I=\{i : x_i=\hat{x}_{\nu i},\ i=1,\cdots,m\}$

とし，$I=\phi$（空集合）ならストアしない，$I\neq\phi$なら

(8) $x_i = b_{i1}t_1+\cdots+b_{ik}t_k \ , i\in I$（$i$の番号の若い順に並べる）

なる方程式の解 $t_1,\cdots,t_k$（一般には不定解となるが，t_1,t_2 … の順に掃き出し，ピボットは(8)式の若い順番に選ぶという方法で，基準形に変換したときの基底解をとるというルール*に従うものとする）が t_ν に一致すればストアし，そうでなければストアしない。

たとえば表2のレコード② $x=(1100101)$ を $\hat{x}_{13}=(0101110)$ にストアすべきかを判定しよう。

まず $I=\{2,3,5\}$ となり方程式(8)は

(9)
$$\begin{cases} t_2+t_3 = 1 \\ t_2+t_4 = 0 \\ t_1+t_3 = 1 \end{cases}$$

となるが，これにルール＊による掃き出しをほどこして変形すると，

$$(10)\quad \begin{cases} t_2 + t_4 = 0 \\ t_3 + t_4 = 1 \\ t_1 + t_4 = 0 \end{cases}$$

なる基準形が得られる．この基底解は非基底変数 t_4 を0とおいて得られるもので，$(t_1, t_2, t_3, t_4) = (0, 1, 0, 0)$ となるが，これは $t_{13} = (1, 1, 0, 1)$ と一致しないから，② は $\mathfrak{X}_{13}$ にはストアしない．

この方法で⓪，①，②，③ と S の各点にストアした結果が表3の右の部分である．(8) の解を一意に決めるためには，上記のような方法でもよいし，解 t の全体のうちこれを2進数とみて一番若いものを優先するというルールでもよい．以上のものは[1]のものと本質的に同等である．ただし[1]では検索をより容易にするため，上のバケット $\mathfrak{X}_p$ をさらに細かなサブバケットに分割する方法がとられている．

§4 検索アルゴリズム

t 項目の質問に対する検索とは，$a_{i_1}, \cdots, a_{i_t}$ が与えられたとき，

$$(11)\quad x_{i_j} = a_{i_j}\ ,\ j = 1, \cdots, t$$

となるような，ストアされている，レコード $x=(x_1,\cdots,x_m)$（のレコード番号）をすべて取り出すことである．[1]などでは t 項目以下の質問に即答できるために，S を決定する(5)に対応する1次式がどの t 個も1次独立であること（強さ t の条件）を要請している．こうすると(11)に対応する方程式

(12) $$a_{ij}=b_{ij1}t_1+\cdots+b_{ijk}t_k \quad , j\in\{1,\cdots,t\}$$

がつねに解をもつから，この解（ストアの場合と同一のルール＊で基準形の基底解として一意に決める）$(t_1,\cdots,t_k)$ に対応するバケットの中に，(11)をみたすすべてのレコードが含まれている．これが[1]の検索アルゴリズムである．

われわれの S は R に適合しているかわり，強さ t の条件をみたしていない．したがって任意に与えられた t 個の値 a_{ij}（$j=1,\cdots,t$）に対して(12)の解が必ずしも存在しない．しかしその場合は，$\{i_1,\cdots,i_t\}$ を適当にいくつかのグループ $G_1,\cdots,G_\Delta$ にわけ（$\{i_1,\cdots,i_t\}=G_1\cup\cdots\cup G_\Delta$，$G_i,G_j$ は必ずしも disjoint でなくてよい），各 G_i に対して上と同様に検索して（(12)の $j\in G_i$ に対する式が解をもつようになるまで細分する）得たレコードの集合 R_i の共通部分

(13) $$R_1\cap\cdots\cap R_\Delta$$

をとれば，これが求めるものであることは容易にわかる．

このように検索ルールを拡張すれば，九項目と言わず任意数項目に対する検索可能な（必ずしも即答でないが）ファイリングシステムが得られる．

§5 今後の問題

ここでのべた最小ハミング距離法は，GF(2) の場は，リードマラーコードのデコード方式と同等でうまく行くが，一般の GF(q) の場合には必ずしも確立されていないようである．今後の研究に待つ．

またここでは，与えられたデータ R に S をあてはめるのに，表2に示すようなキー $(t_1, \cdots, t_k)$ を勝手に与えたが，キーの順序によって当てはめられた式(5)は大きく変る．実際にはキーはファイル R の中のいくつかの項目 $(x_{j_1}, \cdots, x_{j_k})$ によって決まることが多いので，$(t_1, \cdots, t_k)$ のかわりに，このようなそのファイルに固有のキー $(x_{j_1}, \cdots, x_{j_k})$ を用いることが望ましい．こうして残りの x_j をキーの1次式として，最小ハミング距離法を用いて，表現すればよいのである．

このようなキーが R の中にない場合はどうすべきか．それには，$x_1, \cdots, x_m$ の1次式としていくつかの t_i を構成することが考えられる．つまり

(14) $t_{\nu i} = w_{i1} x_{\nu 1} + \cdots + w_{im} x_{\nu m} + e_{\nu i}$

$\nu = 1, \cdots, n, \quad i = 1, \cdots, k$

とし，$\{ e_{\nu i} : \nu = 1, \cdots, n, \ i = 1, \cdots, k \}$ のハミングウェイトが最小になるように w_{ij} を決めて，$x_1, \cdots, x_m$ の情報を $t_1, \cdots, t_k$ に集約すればよい．これは T 度実数の場での主成分分析に相当するものであるが，[5] この効率よいアルゴリズムも未完成である．

またたとえ上のような意味で R に適合した S が得られたとしても，これが R に無関係に決めた S より低い冗長度を与えるかどうかは，直観的にはうなづけるとしても，理論的な証明はまだなされていない．さし当っては，いくつかの例でシミュレーションによって実験的に確かめてみるのが手始めであろう．

参考文献

[0] F. J. Mac Williams & N. J. A. Sloane "The Theory of Error Correcting Codes" North Holland (1977)

[1] R. C. Bose, C. T. Abraham & S. P. Gosh File Organization of Records for Multiple valued Attributes for Multiattribute

Queries" IBM Research Paper RC-1886 (1967)

[2] R.C. Bose & Gary. G. Koch "The Design of Combinatorial Information on Retrieval Systems for Files with Multivalued Attributes" SIAM J. App. Math. (1969)

[3] 高橋磐郎 "ガロア体の離散データ処理への応用" 数理科学 (1979年11月)

[4] I. Takahashi "Filing Schema by Galois Functions" Information & Control (投稿予定)

[5] 奥野忠一他 "多変量解析法" 日科技連出版 (1971)